

## Development of a Quality Control Programme for steel production: A case study

O.E. Isaac and A.K. Le-ol

Department of Mechanical Engineering, Faculty of Engineering, Rivers University of Science and Technology, Port Harcourt, Rivers State Nigeria

---

**Abstract:** In the steel making process, effective scheduling is needed for improvement of productivity. This paper study a dynamic process with a release time, where the process times of job may change during production process due to uncertainties, the objective is to ensure continuity of the production process and just in time delivery of final products. A solution methodology is developed which combine a model predictive control (MPC) strategy based approach and lagrangian relaxation algorithm. The MPC approach tackle the parallel process scheduling problem, and the rolling horizon approach allows applying lagrangian relaxation algorithm to solve the model of the scheduling problem in a rolling fashion. Computational experiments are carried out comparing the proposed. Method with the passive adjustment method often adopted by some quality control engineers. The result shows that the proposed method yields significantly better results.

**Keywords:** model predictive control, lagrangian relaxation algorithm.

---

### NOMENCLATURE

I	= Set of all jobs, $I = \{1, 2 \dots N\}$ , is the total number of jobs
M	= Total number of identical parallel machines
$p_i$	= Processing time of job $i$
$b_i$	= Starting time of job $i$
$r_i$	= Release time of job $i$
$M_k$	= Number of machines available at time $k$
K	= Total number of time units to be considered
R	= Total number of rolling windows during the whole time horizon
U(t)	= Set of jobs that have not been processed until the decision point $t$
$W_i$	= Weight of job $i$
Sp	= Steel make respond
Rf	= refining furnace
Cf	= Converter furnace
HFS	= Hybrid flow shop
$C_i$	= completion time of job $i$ in a rolling window, $i \in I$ , it is a time point ( $C_{i=k}$ means that the job completes at the end of time unit $k$ )
$C_I^0$	= Completion time of job $i$ in the initial schedule, $i \in I$ $\delta_{ik}$ =1 job $i$ is processed at time unit $k$ , $i \in I$ , $k = 1, 2 \dots, k 0$

---

### I. Introduction

The metal-forming industry is an important link in the manufacturing chain, supplying extrusions, tubes, plates and sheets to many major manufacturing enterprises, including the automobile, aircraft, housing and food services, and beverage industries (Balakrishnan and Brown 1996). Iron and steel production includes several process phases (iron-making, steel-making – continuous casting and steel rolling), and is very extensive in investment and energy consumption. It is also characterized by high-temperature high-weight material flow with complicated technological processes. To accommodate customer requirements for different types of finished products with fluctuating demands, different rolling mills in the steel-rolling phase are designed with sufficient production capacity. Since the steel-making process (SP) phase needs expensive and energy-extensive equipment and runs in a continuous mode, its capacity is usually below the total capacity of the rolling stage. Effective scheduling of SP resources is therefore vital, especially in today's highly competitive global steel market. The steel-making process consists of three stages: steel-making, refining and continuous casting. Each stage further includes parallel machines, as shown in figure.1  
The following is a brief description of the production process.

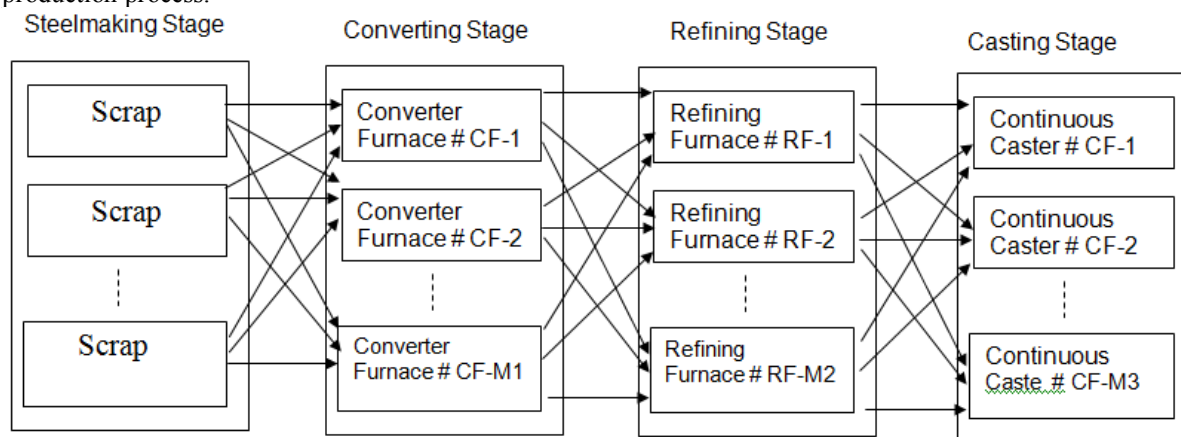
In the steel-making stage, carbon, sulphur, silicon, and other impurity contents of molten iron are reduced to desirable levels by burning with oxygen in a converter (CF) or electric arc furnace. The output from this stage is molten steel with the main alloy elements. The basic unit of steel-making production is a *charge*, which is defined as a 'job' in SP scheduling. It refers to the concurrent smelting in the same converter. The set-up and maintenance times at the steel-making stage are comparatively short. The steel in one charge may be cast into different slabs that are used to produce finished steel products for different customer orders. Charges must be so designed that (1) the orders in the same charge have identical steel grade and gauge; (2) the slab widths for these orders should be within certain limits; (3) delivery dates of the orders are close to each other; and (4) total slab weight ranges from 95% to 100% of the furnace capacity.

The molten steel from the steel-making stage is poured into ladles that are transported by a crane to a refining furnace (RF) for refining. The operation at this stage further refines the chemicals and eliminates impurities in the molten steel or adds the required alloy ingredients. If no RF is available when a new charge arrives, the charge has to wait until one of the RFs becomes available. The waiting time of a charge causes the charge temperature to drop, and reheating is needed, Energy consumption thus grows as the waiting time increases Lixin (2002). The duration of the operation at the refining stage is usually similar to that of the operation at the steel-making stage. After refining, molten steel is poured into a tundish for casting. In the casting stage, molten steel flows down from a hole at the bottom of the tundish into the crystallizer, the input unit, of a continuous caster. The molten steel continuously solidifies into slabs at the bottom of the caster. A sequence of charges that are consecutively cast on the same intermediate ladle and on the same continuous caster using the same crystallizer is called a cast, which is defined as a 'job group' in SP scheduling. Charges in the same cast need to satisfy the following technological constraints.

(1).Release-time related events: (a) Arrival delay of raw materials at upstream operations. (B).Machine breakdown at upstream operation(c)Quality flaw of liquid steel at upstream operation (d) Change of processing speed at upstream operation

(2).Processing – time related events:(a). Change of casting speed (b).Online casting width adjusting (c). Caster stream breakdown (e). Unsatisfactory quality of casting slabs.

The charges in each cast and their sequence are fixed. The diagrams below show the following production process.



SP scheduling is then to decide the schedule of these jobs on the process. Planning and scheduling problems in iron and steel production have not drawn as much attention of the production and operations management researchers as many other industries, such as metal cutting and electronics manufacturing Tang and others. (2001). Because steel-making scheduling with practical constraints is extremely complex, most approaches to SP scheduling treat the problem at three levels (Numoao and Morishita 1991): (1) sub-scheduling, which fulfils the scheduling of individual charge sets; (2) rough scheduling, which merges sub-schedules; and (3) optimal scheduling, which eliminates machine conflicts. (Tang and others 2000), the sub-scheduling and rough scheduling were realized through human-computer inter-action, a nonlinear mathematical model was built for the machine conflict problem, and the model was then converted into a linear programming model for easy solution. Kalagnanam and others. (2000) addressed the problem of satisfying customer orders in an order book with surplus slab inventory before scheduling the production for the remaining orders. The problem was formulated as a bicriteria multiple knapsack problem with additional constraints and was then solved by using a network-based heuristic. Cowling and Rezing (2000) developed a mixed integer programming model and a heuristic for integrated production planning of a steel continuous caster and hot strip mills. Change and others. (2000) studied the lot planning problem to group charges into casts for a continuous slab caster in a integrated steel mill. An integer-programming model and an efficient heuristic were developed the column generation approach combined with a simple round-off scheme. this paper, we develop an integrated model for the SP

scheduling problem including all four SP stages, and present a Lagrangian relaxation method to solve it. The steel-making plant in Nigeria is taken as the research background, and the goal is to generate daily schedules for steelmaking process and help the Nigeria steel company planner to make better short term(for a shift) scheduling decision and also produce a quality steel for its customer.

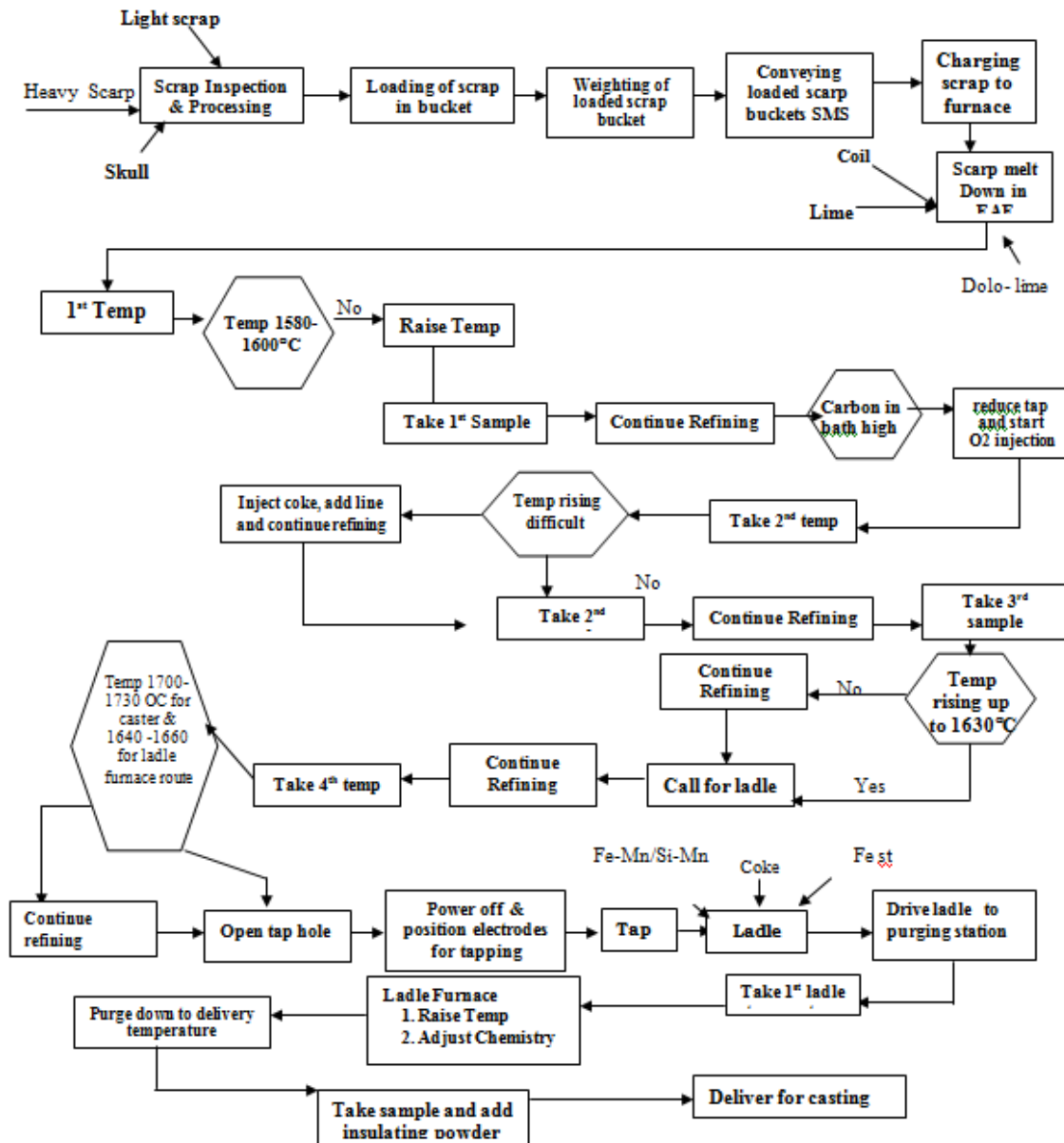


Figure 1.2 Steel Making Process Flow Chart: Scrap Processing to Ladle Treatment

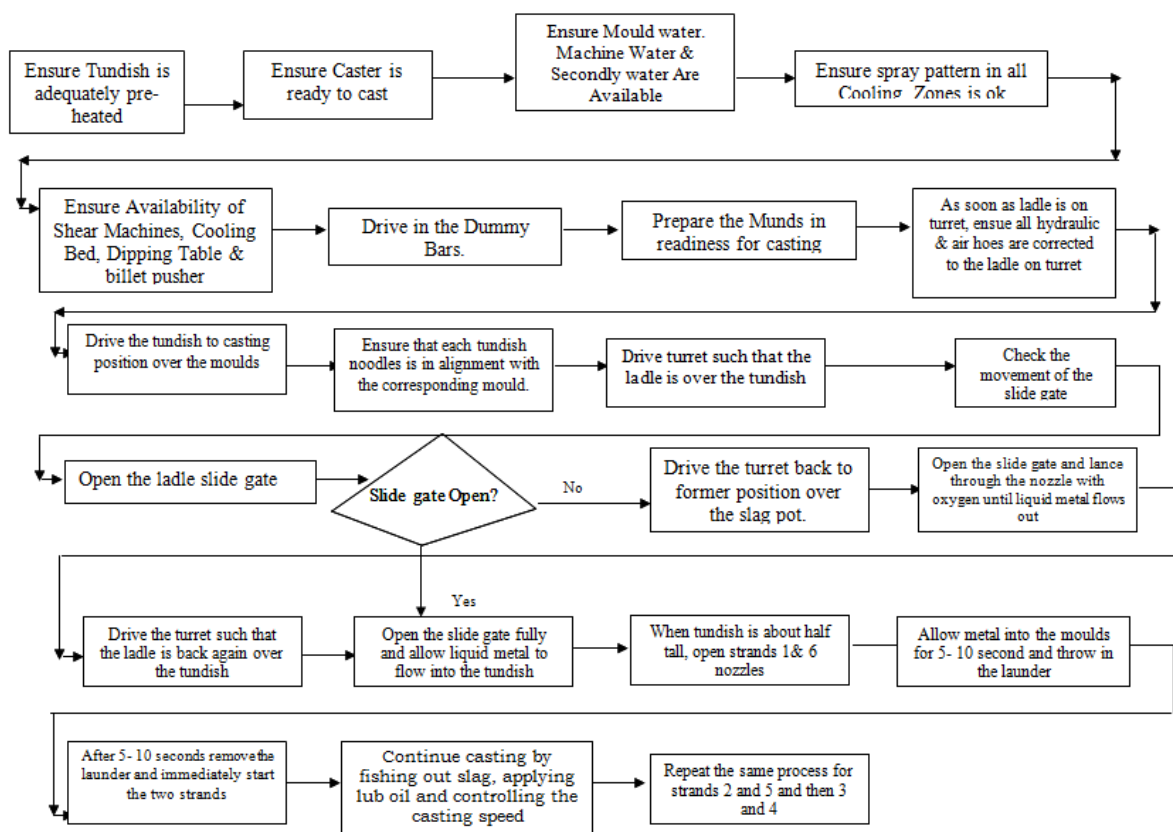


Figure 1.3: Process Flow Chart for Continuous Casting

## II. Mathematical Formulation Of The Problem

### *The problem characteristics and modeling requirements*

After the charges and casts are defined by lot planning, the task of SP scheduling is to determine when and where (on which device) each charge should be processed at each production stage. The following general assumptions usually made in HFS scheduling are also viable for this problem.

- All charges follow the same process route: Steel-making, refining, and then continuous casting. At each stage, a charge can be processed on any one of the machines at that stage, and the parallel machines at that stage are identical.
- A machine can process at most one job at a time.
- A job can be processed on at most one machine at any time.
- Job processing is non-pre-emptive.

However, it is clear from the introduction that the problem has some special features as compared with the scheduling in a general HFS. These features are summarized as follows.

- The number of charges to be scheduled within a shift is not large. However, each charge contains more than 100 tons of molten steel at high temperature. Transfer times between stages needs to be considered.
- Some specified charges (a cast) must be processed as a group on the same caster and there are precedence constraints among the charges within a group at the casting stage.
- Set-up and removal times are required, respectively, before and after a whole cast is processed on a caster. These times are separated from the processing times.
- Idle time on the caster between charges within a cast (cast break) is undesirable and involves cost.
- Waiting time of charges between the processing at different stages causes a temperature drop and results in cost for heating.
- Both earliness and tardiness on job completion lead to cost, e.g. for inventory or compensation to customers.

### SOLUTION FRAMEWORK

Model predictive control (MPC) is an advanced control technique, which has been widely used in the process industries. The usefulness of MPC is that it can enable the system to make the appropriate changes to counteract foreseen disturbances effect in advance by introducing a feed forward effect based on a model and forecast of future behavior. Due to this feature, predictive control can reduce the influence of parameters of the

scheduling object and the uncertainty of circumstance effectively. As a tactical decision policy, MPC can offer a good decision framework that best fits the dynamic problem with disturbances. The logic of the MPC will be applied to model and tackle the dynamic scheduling with disturbances in this paper.

The appeal of the proposed method over traditional approaches to the dynamic scheduling problem in the ever-changing environment includes the following: (1) Under an MPC based framework, a rolling horizon approach can handle the disturbances in the dynamic environment. (2) The application of Lagrangian relaxation embedded into the rolling horizon strategy tackles scheduling problems. The problem under study is first solved based on available information at the beginning to generate an initial schedule (past output). The counteraction of new outputs and reference trajectory is selected to be the input of the controller in or keep the process as close as possible to the reference trajectory. The established model is optimized by the classical methodology (Lagrangian relaxation) to obtain the future inputs or control signals based on the past and current values. Once the control decisions are obtained, the first control signal is sent to the system while all the next control signals are rejected. Then, the horizon advances one step to the next rolling window and the solution procedure is iterated until the last rolling window is reached.

### ROLLING HORIZON IMPLEMENTATION

Rolling horizon procedures have been developed for several different problems. Since it optimizes the system in a limited time horizon instead of globally optimizing the system, it is very suitable for the dynamic scheduling problems in varying environment. With this strategy, the complicated dynamic scheduling problem can be solved in a rolling horizon fashion.

The rolling horizon used here is a little different from the common rolling horizon approach. Suppose that the original scheduling horizon  $(o, k)$  is divided into  $R$  time periods with equal length  $T$ . For the practical steel factory, a natural day or shift is generally taken as the scheduling horizon and an initial schedule is made for the whole horizon. Before the start of each time period, job information may change due to disturbances in the production and thus rescheduling is needed using the original schedule as a reference. In the commonly used rolling horizon approach, at each decision point the rescheduling problem would be solved considering a rolling horizon with fixed number of time periods. To take into account all available information, each rescheduling model in the rolling horizon method used here adjusts the schedule for all the remaining time periods in the same day or shift. We call the time interval from the decision point to the end of the scheduling horizon the rolling window. As time progress, the window is shortened by one time period with the starting point of it moving forward while its end is fixed to the end of the (original) scheduling horizon.

**Model for a Rolling Window.** The following notation is used to present the mathematical model. Terms are defined in the Nomenclature section.

$$PO : \min imize \sum_{i=1}^N w_i C_i + \sum_{i=1}^N (b_i - r_i)^2 \quad (1)$$

$$pl : \min imize \sum_{i \in U(t)} w_i C_i + \sum_{i \in U(t)} (b_i - r_i)^2 + \sum_{i \in U(t)} (C_i - C_i^0)^2 \quad (2)$$

$$s.t \quad r_i \leq b_i \quad i \in U(t) \quad (3)$$

$$\sum_{i \in U(t)} \partial_{ik} \leq M_k \quad k \in [aT, K] \quad (4)$$

$$\sum_{k=t}^k \partial_{ik} = p_i, \quad i \in U(t) \quad (5)$$

$$k \partial_{ik} \leq C_i, \quad i \in U(t); k \in [aT, K] \quad (6)$$

$$C_i - P_i + 1 \leq k + K (1 - \partial_{ik}), \quad i \in U(t); k \in (aT, K) \quad (7)$$

$$\delta_{ik} \in \{0,1\}, \quad i \in U(t); k \in [aT, K] \quad (8)$$

In this paper, parallel process scheduling with release times in the dynamic environment is considered. A mixed integer nonlinear programming model is presented for the scheduling (rescheduling) problem in the rolling horizon.

Initial schedule based on estimated parameters for the whole scheduling horizon is still useful to provide the internal planners and external customers with an expected completion time of their orders. At the beginning time, the problem with objective 1 is first solved based on available information and then an initial feasible solution is obtained  $(C_i^0, \quad i \in I)$ . A new schedule is generated periodically with objective 2 subjected to the newly obtained production information during each rolling window  $[aT, K]$ . Each scheduling

model corresponds to a fixed value of  $a$ . Objective 2 consists of minimizing following three terms: (a) total weighted completion times of jobs, which is the goal in the deterministic environment; (b) energy consumption of jobs caused by waiting to be processed after they are released (Due to high temperature of the products during the production process, a waiting penalty is introduced to the objective in order to reduce energy consumption caused by waiting for processing. Here, the energy consumption of a job is regarded as a quadratic function the job's waiting time); (c) total deviation of actual job completion times from those in the initial schedule. The deviation chosen may reduce the effect of the system arising from dynamic disturbances in the process of scheduling as much as possible. Constraints 3 reflect that a job can be processed only after its release times. Constraints 4 are the machine capacity constraints. They indicate that all these machine requirements are satisfied with the number of available machines at that time. Machine capacity cannot be exceeded at each time unit. Constraints 5 ensure that the time intervals for which a job occupies one of the machines must equal to the required amount of time for processing. Constraints 6 ensure that a job occupies one of the machines until the processing is finished. Constraints 7 ensure that a job starts occupying a machine from its beginning time. Constraints 5-7 together represent the time intervals for which a job is processed on a machine. And finally,

Based on the MPC framework, the model established for scheduling problem corresponding to each rolling window is different from the most existing traditional scheduling models in three aspects. (1) The waiting penalty is introduced to the objective for saving energy. (2) The model can tackle the dynamic scheduling with disturbances. (3) The scheduling deviation is taken as the objective to reduce the inconvenience cost caused by uncertainties.

### LAGRANGIAN RELAXATION ALGORITHM

Lagrangian relaxation is an efficient optimization method. The major advantage is that it can provide a lower bound for problems (fisher1981). It has been applied to a wide range of scheduling problems such as the parallel machine scheduling problem (Luh and others 1990) or the job shop scheduling problem. Lagrangian relaxation is presented to solve the model corresponding to each rolling window in this work. The coupling constraints are relaxed and embedded as a penalty term multiplied by Lagrange multipliers in the objective function, therefore leading to a relaxed version of the primal model, which can be decomposed into several independent job-level sub problems to reduce the problem size. It is sufficient to obtain a near-optimal schedule with quantifiable performance within a reasonable computation time. The feasible solutions to Lagrangian relaxation at the beginning is an initial solution using objective 1. During the future rolling window, the scheduling problem is rescheduled again based on the update information in the window and scheduling result is adjusted.

The job-level decomposition of the model is introduced by relaxing constraints 4 with Lagrangian multipliers ( $uk$ ). The first solving LR problem is denoted by  $LO$ , and the LR problem solved at each rolling window is denoted by  $L1$ , respectively.

$$\begin{aligned}
 (LO) \quad & \min_{(C_I)} \sum_{i=1}^N w_i C_i + \sum_{i=1}^N (b_i - r_i)^2 + \sum_{k=1}^k uk \left( \sum_i \delta_{ik} - Mk \right) \\
 & = \min_{c_i} \sum_{i=1}^N \left[ w_i C_i + (b_i - r_i)^2 + \sum_{k=1}^k uk \delta_{ik} \right] - \sum_{k=1}^k uk M_k \\
 & = \min_{c_i} \sum_{i=1}^N \left[ w_i C_i + (C_i - P_i + 1 - r_i)^2 + \sum_{k=1}^k uk \delta_{ik} \right] - \sum_{k=1}^k uk M_k \tag{9}
 \end{aligned}$$

$$\begin{aligned}
 (L1) \quad & \min_{c_i} \sum_{i \in U(t)} w_i C_i + \sum_{i \in U(t)} (b_i - r_i)^2 + \\
 & \sum_{i \in U(t)} \left[ w_i C_i + \sum_{k=t}^k uk \left( \sum_{i \in U(t)} \delta_{ik} - Mk \right) \right] \\
 & = \min_{c_i} \sum_{i \in U(t)} \left[ w_i C_i + (b_i - r_i)^2 + (C_i - C_i^0)^2 + \sum_{k=t}^k uk \delta_{ik} \right] - \sum_{k=t}^k uk M_k \\
 & = \min_{c_i} \sum_{i \in U(t)} [w_i C_i + (C_i - p_i + 1 - r_i)^2 +
 \end{aligned}$$

$$(c_i - c_i^o)^2 + \sum_{k=t}^k uk\delta_{ik} - \sum_{k=t}^k ukMk \quad (10)$$

The job – level subproblems are solved optimally by enumeration. The lower bound together with the upper bound and the deviation in the linking constraints is used to update the Lagrange multipliers according to a subgradient updating procedure. In our algorithm, the subgradient optimization method is used to update the multipliers (uk) at each iteration given by  $u^n + 1 = u^n + \alpha^n g(u^n)$ , where u is a vector of multipliers (uk), n is the iteration index, and g (u) the subgradient of L equal to  $(\sum_i^n \delta_{ik} - M_k)$ . And  $\alpha^n$  is the step size at the nth iteration defined as  $\alpha^n = \lambda(L^* - L^n) / (\|g(u^n)\|^2)$ ,  $0 < \lambda < 2$ , where  $L^*$  is estimated by the best feasible objective value found so far and  $L^n$  is the dual value at the n<sup>th</sup> iteration.

### III. Computational Experiment

To test the performance of the method and to study the characteristics of the solutions, a computational experiment has been carried out on randomly generated problem instances, which were designed to reflect practical situations in iron and steel industries. The method was implemented by using visual basic .net 2008 and the experiment was carried out on a HP pavilion dv5 notebook 2.27GHZ PC.

#### LAGRANGIAN ALGORITHM

##### The Algorithm Procedure:

The following are the step to step algorithm process of evaluating the objective function of Lagrangian relaxation from steel production scheduling

Step1: Declaration of variables used in the program

Step2: Input the values of N, which is the array size.

Step3: Input the values of  $W_i, C_i, U_k, \delta_{ik}$  and  $M_k$  to compute the values of  $b_i, r_i, (b_i - r_i)^2, W_i, C_i, (\delta_{ik} - M_k)$

Step 4: Calculate the values of array  $L_o$  and  $P_o$  using the formula:

$$L_o = \frac{\min}{(C_i)} \sum_{i=1}^N w_i C_i + \sum_{i=1}^N (b_i - r_i)^2 + \sum_{k=1}^K uk \left( \sum_{i=1}^N \delta_{ik} - M_k \right)$$

$$P_o = \text{Minimize} \sum_{i=1}^N w_i C_i + \sum_{i=1}^N (b_i r_i)^2$$

Step 5: Click transfer to send the values of  $L_o$  and  $P_o$  to the phase II of the algorithm is as follows:

Step 1: Declaration of all variable used in phase II

Step 2: Input the value of N which is the size of the array

Step 3: Enter the number of jobs machines and number of rolling window which makes up the 'problem structure'

Step 4: Enter the values of the running time in seconds

Step 5: Initialize of sum  $l_o = 0$ , sum  $P_o = 0$ , sum  $lmp = 0$ , sum  $rt = 0$

Step 6: Calculate the improvement (%) using the formular: improvement =  $(L_o - P_o) / P_o \times 100\%$

Step 7: Compute the sum of  $L_o, P_o$  Improvement and running time.

Step 8: Calculate the averages of  $L_o, P_o$ , Improvement and running time.

Step 9: Plot the graph of improvement against number of machines used in processing

Step 10: End program

### IV. Generation Of Problem Instances

To generate representative problem instances, we examined the actual production data from a Steel Production Firm in Nigeria, which is the largest and most advanced iron and steel enterprise in Nigeria, its annual production is over 18 million tons of steel, and its auto sheet accounts for more than 6% of the domestic market share. In the steel-making plant in Nigeria every workday is divided into three shifts. Planners need to carry out SP scheduling in every shift for the next shift. The SP schedule generally includes about 5-7 casts for each workday, and a cast consists of 3-5 charges subject to technological constraint. The maximum number of charges to be scheduled in each workday is about 35 (about 12 per shift). Based on the above, the number of

charges to be scheduled is set to be 12 for each instance. Since the minimum scheduling time unit in the iron and steel plant is in an exact number of minutes, the minute is taken as the basic time unit. The planning horizon is set to be 480 minutes, as this study intends to solve the SP scheduling problem for an eight-hour shift. Two other parameters are chosen to represent the problem structure as described below:

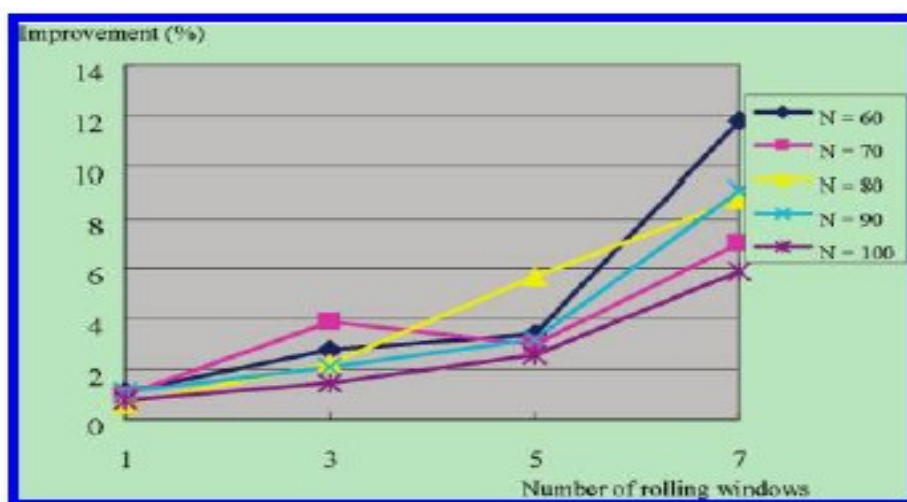
1. The number of casts is set to vary at three levels: 3, 4, and 6.
2. The number of machines at each stage is set to vary at three levels: 3, 4, and 5.

In order to reduce experimental cases, it is assumed that every stage has the same number of machines. However, our method presented here can deal with practical problems including different numbers of machines for different stages.

The combination of parameter levels gives nine problem scenarios, and for each scenario, ten different problem instances were randomly generated. Thus, a total of 90 problem instances were used in the experiment. According to practical data of the steel-making plant of Delta steel. The proposed method with the PAM. The PAM is an often used method by human schedulers for scheduling in systems with uncertainties, and it considers disturbances of the jobs during each rolling window. At each rolling window, the newest data and information is first updated. The jobs after being disturbed are compulsorily moved to be processed at the available time of the machines. And, the subsequent jobs which follow the disturbed jobs in the initial schedule need to be adjusted respectively. This procedure stops until all jobs are scheduled in the last rolling window.

**Figure 4.1 Effect of the number of rolling windows on algorithm performance**  
(Disturbance on release time of jobs).

Problem No.	Problem structures (Jobs x machines x No. of Rolling windows)	LO	PO	Improvement (%)	Running time (s)
1	60 x 3 x 1	130663.0	132158.4	1.14	0.06
2	60 x 3 x 3	121866.2	125216.3	2.75	0.22
3	60 x 3 x 5	127445.6	131769.9	3.39	2.19
4	60 x 3 x 7	132567.1	148176.2	11.77	0.80
5	70 x 3 x 1	160814.0	162358.9	0.96	0.08
6	70 x 3 x 3	173716.7	180473.1	3.89	0.32
7	70 x 3 x 5	174505.2	179716.8	2.99	3.14
8	70 x 3 x 7	173250.9	185379.5	7.00	4.30
9	80 x 3 x 1	216120.7	217469.1	0.62	0.11
10	80 x 3 x 3	204384.2	208882.8	2.20	0.42
Average		161533.4	167160	3.671	1.164



**Figure 4.1 Effect of the number of rolling windows on algorithm performance**  
(disturbance on release time of jobs).

For the experiments, the objective value by using PAM is denoted as PO. And the objective value by using the proposed approach is denoted as LO. The column improvement is calculated by  $(PO - LO)/LO \times 100\%$ . To guarantee optimal solution

In this section, extensive experiments are conducted based on different parameter configurations to analyze the performance of the propose approach. For the tests, the number of jobs ranges from 20 to 100. The number of machines varies at three levels (3, 4, and 5). The number of rolling windows varies at four levels (2, 4, 6, and 8). The data for a job  $i$  are randomly generated as follows. An integer processing time  $P_i$  is generated



from the uniform distribution [5, 15], and an integer release time  $ri$  are generated from the uniform distribution  $[1, \frac{1}{2} \sum_i^N Pi]$ . An integer weight  $wi$  is generated from the uniform distribution [10, 15].

Disturbances are represented in a stochastic fashion. The entire time horizon is divided into a number of rolling windows with dynamic events on unscheduled jobs. As done in this paper, we study two kinds of dynamic events occurring during each rolling window based on the practical production of the steel complex, including variations of the process times and the release times on jobs. The disturbance scope for the processing time of each job is generated from the uniform distribution  $[0, \max(Pi(ri - t)/10, 20)]$ . The disturbance scope for the release time of each job is generated from the uniform distribution  $[0, \max(ri - t, 20))$ , where  $t$  is the decision point of the current rolling window. The combination of the above parameter levels generates 108 different problem scenarios, for each of which 10 different problem instances have been tested.

A periodic rolling horizon scheduling strategy is adopted in this paper where a natural day or shift is taken as the whole time horizon. In order to illustrate the performance of the proposed method with respect to the different number of jobs, we carry out the computational experiments and the results are presented in the number of rolling windows is set to different values to test its impact on solution quality. Illustrate the solution performance against different rolling window sizes and different numbers of jobs for disturbance on processing times of jobs and release times of jobs, respectively. There is a clear impact of the number of rolling windows on solution quality. As the number of rolling windows increases, improvement becomes larger gradually which means that the solution quality of the proposed approach becomes better steadily, compared to the solution of the PAM. As the number of rolling window increases, the disturbances in the dynamic scheduling environment can be measured and handled in a timely manner, which results in improvement on the quality of the solutions, although computation time increases.

The experiments are carried out based on one kinds of situations including the disturbance of the jobs' processing times and the disturbances of the jobs' release times. Upon the basis of results presented in Table 4.1, we can make the following observations about our methods.

## V. Conclusion

When the number of jobs and the number of rolling windows are fixed, the number of machines increase in most cases. This is consistent with the intuition that for a fixed number of jobs and rolling windows, when the number of machines increases, there is more choice of machines for each job and the problem becomes easier to solve. From the column of improvement in the tables, it was observed that our proposed method greatly outperforms the passive adjustment method, which shows that the proposed method yield significantly better result with good improvement.

## References

- [1]. Balakrishnan, A. and Brown, S., (1996), Process planning for aluminium tubes: an engineering-operations perspective. *Operations Research*, **44**, 7-20.
- [2]. Cowling, P. and Rezig, W., (2000), Integration of continuous caster and hot strip mill planning for steel production. *Journal of Scheduling*, **3**, 185-208.
- [3]. Chang, S. Y., Chang, M. R. and Hong, Y. S., (2000), A lot grouping algorithm for a continuous slab caster in an integrated steel mill. *Production Planning & Control*, **11**, 363-368.
- [4]. Fisher, M. L., (1981), Lagrangian relaxation method for solving integer programming. *Management Science*, **27** (1), 221-218.
- [5]. Guinet, A. G. P. and Solomon, M. M., (1996), scheduling hybrid flow shops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*, **34**, 1643-1654.
- [6]. Gupta, J.N.D., (1988), Two-stage hybrid flowshop scheduling problem. *Journal of Operational Research Society*, **39**, 359-364.
- [7]. Gupta, J.N.D. and Tunc, E. A., (1998), Minimizing tardy jobs in a two-stage hybrid Flowshop. *International Journal of Production Research*, **36**, 2397-2417
- [8]. Hoitomt, D. J., Luh, P. B., Max, E. and Patipati, K., (1993), A practical approach to jobshop scheduling. *IEEE Transactions on Robotics and Automation*, **9**(1), 1-13.
- [9]. Kalagnanam, J. R., Dawande, M. W., Trumbo, M. and Lee, H. S., (2000), Surplus inventory matching problem in the process industry. *Operations Research*, **48**, 505-516.
- [10]. Luh, P.B.; Hoitomt, D. J.; Max, E.; Pattipati, K.P. (1990), Scheduling generation and reconfiguration for parallel machines. *IEEE Trans. Robotic. Autom* 6 (6), 687-696
- [11]. Lixin T, Peter.B, Luh, Jijin L, and Lei F., (2002).steel making using lagrangian relaxation journal of international Production Research **40**,1,55-70
- [12]. Numao, M. and Morishita, S. I., (1991), Cooperative scheduling and its application to steelmaking processes. *IEEE Transactions on Industrial Electronics*, **38**, 150-155.
- [13]. Tang, L. X., Liu, J. Y., Rong, A. Y. and Yang, Z. H., (2001), A review of integrated planning & scheduling systems and methods for integrated steel production. *European Journal of Operational Research*, **133**, 1-20.
- [14]. Tang, L. X., Liu, J. Y., Rong, A. Y. and Yang, Z. H., (2000), A mathematical programming model for scheduling steelmaking-continuous casting production scheduling. *European Journal of Operational Research*, **120**, 423-435.
- [15]. Rajendran, C. and Chaudhuri, D., (1992), A multi-stage parallel-processor flow shop problem with minimum Flowtime. *European Journal of Operational Research*, **57**, 111-122.